Homework 5

# Question 2

import pyspark.sql.functions as fc

# Load Data to dataframes

country\_df = spark.read.json("country.json")

city\_df = spark.read.json("city.json")

country\_lang\_df = spark.read.json("countrylanguage.json")

aqi\_df = spark.read.csv("aqi.csv", header=True, inferSchema=True)

## A

### I

joined\_df = aqi\_df.join(country\_df, aqi\_df.country == country\_df.Name)

result\_df = joined\_df.select("Name").distinct().sort("Name")

result\_df.show(truncate=False)

+----------------------+

|Name |

+----------------------+

|Albania |

|Algeria |

|Andorra |

|Angola |

|Argentina |

|Armenia |

|Australia |

|Austria |

|Azerbaijan |

|Bahrain |

|Bangladesh |

|Belarus |

|Belgium |

|Belize |

|Bermuda |

|Bolivia |

|Bosnia and Herzegovina|

|Brazil |

|Brunei |

|Bulgaria |

+----------------------+

only showing top 20 rows

### II

country\_set = country\_df.select("Name").distinct()

aqi\_set = aqi\_df.select("country").distinct()

result\_set = country\_set.intersect(aqi\_set).sort("Name")

result\_set.show(truncate=False)

+----------------------+

|Name |

+----------------------+

|Albania |

|Algeria |

|Andorra |

|Angola |

|Argentina |

|Armenia |

|Australia |

|Austria |

|Azerbaijan |

|Bahrain |

|Bangladesh |

|Belarus |

|Belgium |

|Belize |

|Bermuda |

|Bolivia |

|Bosnia and Herzegovina|

|Brazil |

|Brunei |

|Bulgaria |

+----------------------+

only showing top 20 rows

## B

### I

joined\_df = aqi\_df.join(country\_df, aqi\_df.country == country\_df.Name, "left\_anti")

result\_df = joined\_df.select("country").distinct().sort("country")

result\_df.show(truncate=False)

+----------------------------------------------------+

|country |

+----------------------------------------------------+

|Ivory Coast |

|Jersey |

|Kazakhstan |

|Kosovo |

|Montenegro |

|Palestinian Territory |

|Reunion |

|Russia |

|Serbia |

|United Kingdom of Great Britain and Northern Ireland|

|United States of America |

|Vatican |

+----------------------------------------------------+

### II

country\_set = country\_df.select("Name").distinct()

aqi\_set = aqi\_df.select("country").distinct()

result\_set = aqi\_set.subtract(country\_set).sort("country")

result\_set.show(truncate=False)

+----------------------------------------------------+

|country |

+----------------------------------------------------+

|Ivory Coast |

|Jersey |

|Kazakhstan |

|Kosovo |

|Montenegro |

|Palestinian Territory |

|Reunion |

|Russia |

|Serbia |

|United Kingdom of Great Britain and Northern Ireland|

|United States of America |

|Vatican |

+----------------------------------------------------+

## C

### I

joined\_df = country\_df.join(aqi\_df, aqi\_df.country == country\_df.Name, "left\_anti")

result\_df = joined\_df.select("Name").distinct().sort("Name")

result\_df.show(truncate=False)

+-------------------------------------+

|Name |

+-------------------------------------+

|Afghanistan |

|American Samoa |

|Anguilla |

|Antarctica |

|Antigua and Barbuda |

|Aruba |

|Bahamas |

|Barbados |

|Benin |

|Bhutan |

|Botswana |

|Bouvet Island |

|British Indian Ocean Territory |

|Burundi |

|Cameroon |

|Christmas Island |

|Cocos (Keeling) Islands |

|Comoros |

|Congo |

|Congo, The Democratic Republic of the|

+-------------------------------------+

only showing top 20 rows

### II

country\_set = country\_df.select("Name").distinct()

aqi\_set = aqi\_df.select("country").distinct()

result\_set = country\_set.subtract(aqi\_set).sort("Name")

result\_set.show(truncate=False)

+-------------------------------------+

|Name |

+-------------------------------------+

|Afghanistan |

|American Samoa |

|Anguilla |

|Antarctica |

|Antigua and Barbuda |

|Aruba |

|Bahamas |

|Barbados |

|Benin |

|Bhutan |

|Botswana |

|Bouvet Island |

|British Indian Ocean Territory |

|Burundi |

|Cameroon |

|Christmas Island |

|Cocos (Keeling) Islands |

|Comoros |

|Congo |

|Congo, The Democratic Republic of the|

+-------------------------------------+

only showing top 20 rows

## D

aqi\_df = aqi\_df.withColumn('year', fc.year('date'))

aqi\_df = aqi\_df.withColumn('month', fc.month('date'))

aqi\_aug\_2022\_df = aqi\_df.filter(

(aqi\_df['year'] == 2022) & (aqi\_df['month'] == 8)

)

aqi\_aug\_2022\_df = aqi\_aug\_2022\_df.groupBy('status')

aqi\_aug\_2022\_df = aqi\_aug\_2022\_df.agg(fc.avg('value').alias('avg\_value'))

aqi\_aug\_2022\_df = aqi\_aug\_2022\_df.filter(fc.count('value') >= 100)

aqi\_aug\_2022\_df = aqi\_aug\_2022\_df.sort("avg\_value")

aqi\_aug\_2022\_df.show(truncate=False)

+------------------------------+------------------+

|status |avg\_value |

+------------------------------+------------------+

|Good |27.929097605893187|

|Moderate |71.38070175438597 |

|Unhealthy for Sensitive Groups|122.953125 |

|Unhealthy |167.9704433497537 |

+------------------------------+------------------+